

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Bakalářská práce

2011

Jakub Ribka

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practise in the
Company

2011

Jakub Ribka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jakub Ribka**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Tieto Czech s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Gaura**

Konzultant bakalářské práce: Bc. Ondřej Kvasnovský

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Poděkování

Na tomto místě bych rád poděkoval všem zaměstnancům firmy Tieto Czech s.r.o., zvláště pak svému nadřízenému Bc. Ondřeji Kvasnovskému, který mě začlenil do týmu a po dobu trvání odborné praxe vždy s ochotou poradil.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 16. března 2011

.....

Podpis

„Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.“

V Ostravě dne 16. března 2011

.....

Podpis

Abstrakt

Tato bakalářská práce se zabývá popisem odborné praxe vykonávané ve firmě Tieto Czech s.r.o. Pracoval jsem na pozici Java developer a po téměř celou dobu této praxe jsem pracoval na aplikaci Quiz, která má sloužit pro testování uchazečů o práci ve firmě Tieto Czech s.r.o. a taktéž pro srovnání již stávajících zaměstnanců a jejich vědomostí z různých oddělení. Na této aplikaci jsem se podílel vývojem, implementací a návrhem.

Klíčová slova

Java, MySQL, Hibernate, Vaadin

Abstract

This bachelor thesis deals with the description of the technical practice performed in the Tieto Czech s.r.o. I was working as Java developer and almost the entire duration of this practice I was working on the application Quiz, which is used for testing applicants for a job in the company of these and also for comparison of current employees and their knowledge of various departments. I was involved in the development, implementation and design on this application.

Key words

Java, MySQL, Hibernate, Vaadin,

Seznam použitých symbolů a zkratek

SVN – Subversion

DAIS – Databázové a informační systémy

TIS – Tvorba informačních systémů

SQL – Sturctured query langure

MySQL – Databázový systém využívající jazyka SQL

HR – Human resources

CSS – Cascading style sheet

PDF – Portable dokument format

DOC – Dokument typu Word

PJ1 – Programovací jazyky 1

JAT – Java technologie

UPR – Úvod do programování

TZD – Teorie zpracování dat

SWS – Softwarové inženýrství

HTML - HyperText Markup Language

LDAP - Stromová struktura záznamů

Obsah

1	Úvod.....	1
2	Odborné zaměření firmy.....	2
2.1	Firma Tieto Czech s.r.o.....	2
2.2	Pracovní pozice ve firmě	2
3	Zadané úkoly v průběhu praxe.....	3
3.1	Přiřazení k projektu Quiz	3
3.2	Analýza aplikace	3
3.3	Nová funkcionálita	4
3.4	Vývoj aplikace	4
3.5	Testování aplikace.....	4
4	Postup řešení zadaných úkolů.....	5
4.1	Přiřazení k projektu Quiz	5
4.2	Analýza aplikace	5
4.2.1	Run test	7
4.2.2	View Result	8
4.2.3	Admin	9
4.2.4	Role v systému	16
4.3	Nová funkcionálita	16
4.4	Vývoj aplikace	17
4.4.1	Export výsledků do Wordu a PDF	17
4.4.2	Dynamické testování.....	19
4.4.3	Editace otázek.....	19
4.4.4	Autentifikace pomocí LDAP	20
4.4.5	Vytváření a editace Uživatelů.....	20
4.5	Testování aplikace.....	20
5	Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.....	21
6	Znalosti či dovednosti scházející studentovi v průběhu odborné praxe	22
7	Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.....	23
8	Seznam obrázků	24
9	Literatura	25

1 Úvod

Jako jeden z mála vysokoškolských studentů jsem měl tu výhodu, že jsem si jako ekvivalent k bakalářské práci mohl zvolit odbornou praxi. Tato možnost mne ihned zaujala, i když jsem o této praxi neměl žádné informace ani reference. V seznamu firem, nabízejících odbornou praxi, se mi zalíbila firma Tieto Czech s.r.o. a pozice vývojáře Java aplikací. Do firmy jsem proto zaslal životopis a následně jsem byl pozván na pohovor. Jelikož zájem o tuto firmu a danou pozici byl veliký, měl jsem obavy, že neuspěji. Asi týden po absolvování vstupního pohovoru mi bylo toto místo nabídnuto. Měl jsem opravdu radost, že si mě firma Tieto Czech s.r.o. vybrala a mohl jsem nastoupit. Byl jsem seznámen s chodem firmy, prošel úvodní školení a mohl jsem se zapojit do práce.

Tato bakalářská práce přiblíží čtenáři vývoj webové aplikace Quiz, která je zaměřena na testování odborných znalostí uživatelů této aplikace. V první části této práce si podrobně přiblížíme jednotlivé funkce této aplikace a v druhé se zaměříme na samotný vývoj aplikace Quiz.

2 Odborné zaměření firmy

2.1 Firma Tieto Czech s.r.o.

Firma Tieto Czech s.r.o. je nadnárodní korporace, která se zaměřuje na komplexní řešení v oblasti IT služeb. V současnosti zaměstnává firma tisíce IT specialistů, kteří se podílí na vývoji produktů šitých na míru širokému okruhu zákazníků. Kromě poskytování IT služeb se tato společnost orientuje také na výzkum v oblasti podnikání, tzn., že se snaží zajistit co nejefektivnější způsoby rozvoje a vývoje produktů. Další ze služeb poskytovaných firmou Tieto Czech s.r.o. je outsourcing a v neposlední řadě taky cloud computing.

2.2 Pracovní pozice ve firmě

Firma Tieto Czech s.r.o. nabízela studentům celou řadu pozic. Mě nejvíce zaujala pozice vývoje Java aplikací, ve firmě jsem tedy nastoupil na místo Java developera. Byl jsem přiřazen na pozici vývoje webové aplikace postavené na technologii J2EE. Moje pozice vyžadovala návrh, implementaci a opravu chyb, které se projevíly během testování. Byl jsem součástí tří a později pěti členného týmu programátorů (rovněž studentů na odborné praxi). Mým úkolem byla také kompletní správa zdrojů (databáze, SVN, server, atd.). Zadané úkoly jsem se snažil plnit tak, aby výsledná aplikace byla co nejkvalitnější, a to jak vzhledově, tak z pohledu funkcionality.

3 Zadané úkoly v průběhu praxe

3.1 Přiřazení k projektu Quiz

Po nástupu do firmy Tieto Czech s.r.o., jsem bezprostředně začal pracovat na částečně rozpracované interní aplikaci Quiz, která po dokončení měla sloužit pro potřebu firmy Tieto Czech s.r.o. Na vývoji této aplikace se podílel celkem tři členný tým lidí. Všichni členové týmu byli rovněž studenti Vysoké školy Báňské a ve firmě Tieto Czech s.r.o. vykonávali odbornou bakalářskou praxi. Jednotlivé úkoly a požadavky při vývoji této aplikace nám zadával pan Bc. Ondřej Kvasnovský, který byl rovněž našim přímým nadřízeným. Původně měla být aplikace Quiz jednoduchou aplikací, které měla být dokončena v horizontu dvou až tří měsíců. Tento plán se ale po našem příchodu radikálně změnil a aplikace Quiz narostla do značné míry. Díky nárůstu složitosti této aplikace bylo zhruba v půlce naší odborné praxe rozhodnuto o rozšíření týmu ze tří členů na pět, a to opět studentů na odborné praxi. Na vývoji aplikace jsme oproti původnímu plánu dělali po celou dobu trvání naší odborné praxe, a to víc než stanovených 50 dní (tuto lhůtu jsme značně přesáhli, jelikož nás vývoj velmi zajímal a bavil). Naším nejdůležitějším úkolem bylo aplikaci zdokonalit jak funkčně tak vzhledově.

Účel této aplikace se zásadně od začátku vývoje nezměnil, jen značně stoupla složitost aplikace. Nicméně celá tato aplikace měla sloužit jako testovací prostředí pro příjem nových zaměstnanců a testování znalostí těch stávajících.

Původně jsme měli být po dokončení Quizu přiřazení na další projekt Building utilization, který měl mapovat využití budov a kanceláří pro firmu Tieto Czech s.r.o. Tomuto projektu jsme se věnovali jen krátce a bohužel na něj už nezbyl čas.

3.2 Analýza aplikace

Po nástupu na projekt Quiz nás čekal nelehký úkol, zorientovat se v rozpracované aplikaci, která již v tu dobu byla poměrně rozsáhlá. Celá aplikace byla postavena na jazyce Java, novinkou však pro mě byl webový framework Vaadin, se kterým jsem se dosud nikdy nesetkal. Tento framework umožňuje psát RIA aplikace, tedy aplikace, kdy programátor píše webovou aplikaci jako desktopovou. Nějakou dobu nám ovšem trvalo, než jsme se naučili s tímto frameworkem pracovat.

Další z věcí, se kterou jsem se musel seznámit, byla datová vrstva aplikace, v našem případě aplikace využívá MySQL databázi. A pro mapování je opět použit framework tentokrát Hibernate. S vlastním SQL jsem díky absolvování předmětů TZD, DAIS a TIS neměl velký problém. Úplnou novinkou však pro měl být framework Hibernate, který jsem musel pomocí tutoriálů nastudovat a který mi ze začátku dělal trošku problémy.

3.3 Nová funkcionalita

Po mém příchodu do Tieto Czech s.r.o. a přiřazení k projektu Quiz jsem začal dostávat úkoly od mého přímého nadřízeného, a to jak z pohledu funkcionality, tak co se týká vzhledu.

Po stránce vzhledové šlo o přátelštější prostředí vůči uživateli, zvýšení přehlednosti a zjednodušení ovládání celé aplikace (user-friendly). Větší změny se týkaly přidání nových funkcionalit systému, tyto úkoly jsme oproti původnímu plánu plnili celou dobu konání naší praxe, jelikož se požadavky na výsledný systém stále zvyšovaly.

Sám jsem se snažil, co nejlépe navrhovat změny a snažit se je implementovat do systému tak, aby vzhled byl uživateli co nejpřívětivější a funkcionalita co nejlepší.

3.4 Vývoj aplikace

Největší změnou, na kterou jsem si musel zvyknout, byla práce v týmu, všechny dosavadní aplikace a projekty, které jsem programoval, jsem vyvíjel sám. Chvilí mi trvalo, si na tuto změnu zvyknout, ale firma Tieto Czech s.r.o. je na tento druh vývoje velice dobře připravena a má k dispozici všechen potřebný software. Nejdůležitější věc bylo sdílení aktualizovaného kódu, který ve firmě Tieto Czech s.r.o. probíhá prostřednictvím SVN. Pro psaní zdrojového kódu jsem používal vývojové prostředí Netbeans.

3.5 Testování aplikace

Po několika měsíčním vývoji, kdy projekt dospěl do fáze, kdy jednotlivé bloky byly spojeny ve funkční celek, můj nadřízený rozhodl, že na tuto aplikaci nasadí testery. Byla to zase nová zkušenost, jelikož jsem se musel naučit synchronizovat svou práci s postupným testováním aplikace. Práce na vývoji systému se v této fázi začala měnit. Ubývalo implementování nových funkcionalit a začaly práce na reportovaných chybách. Vždy jsem se snažil svůj kód a výslednou funkcionalitu otestovat sám, ale vždy nebylo možné zajistit bezchybnost a navíc některé chyby šly odhalit jen velmi těžší. Nicméně pro zkušené testery (jako jediní na našem projektu řadoví zaměstnanci Tieto Czech s.r.o.), kteří se zabývají pouze testováním vyvinutého softwaru, to nebyl velký problém.

4 Postup řešení zadaných úkolů

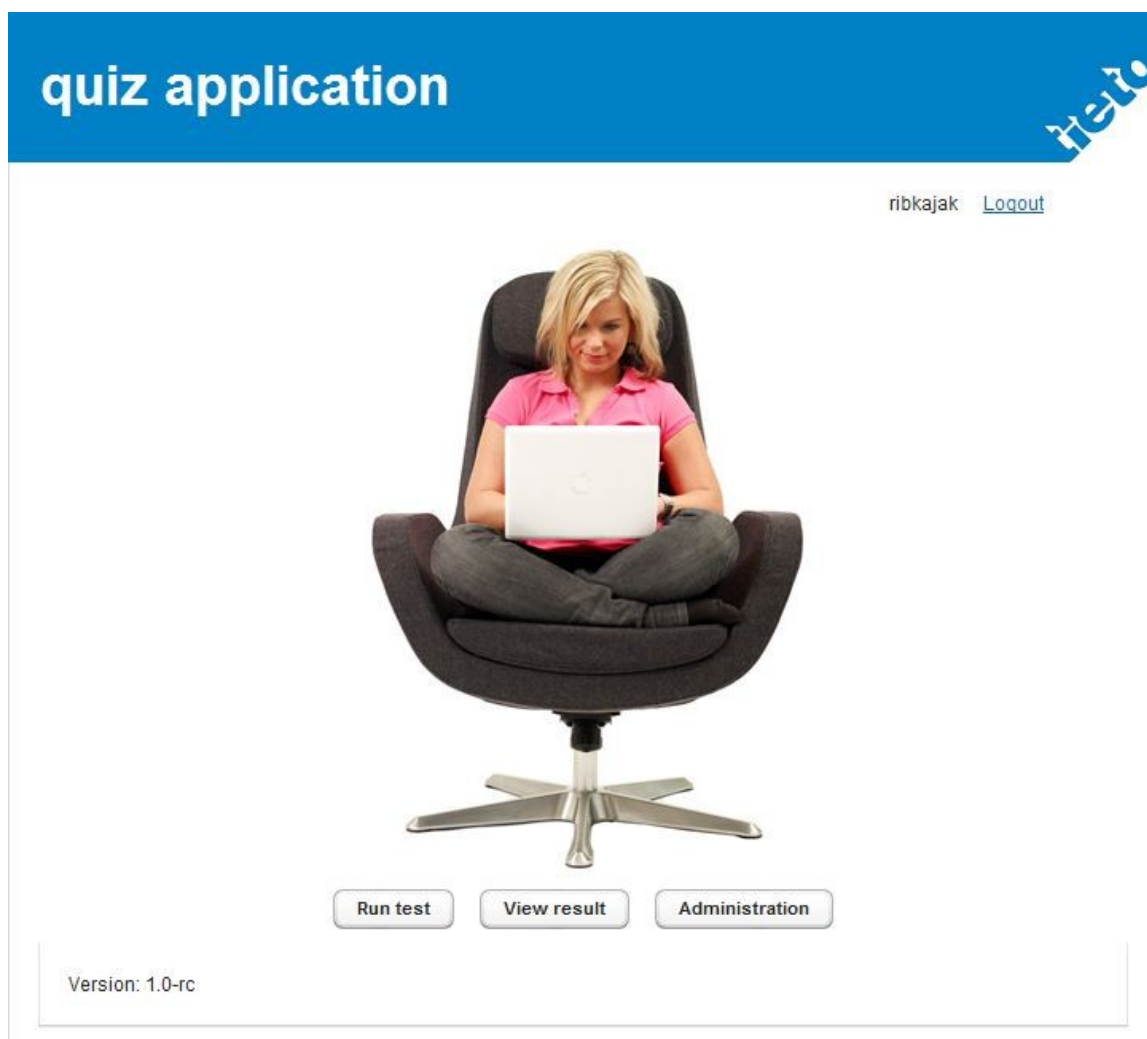
4.1 Přiřazení k projektu Quiz

V prvních týdnech po nástupu do Tieto Czech s.r.o. jsem obdržel všechny přístupové údaje. Stáhnul jsem si veškeré zdrojové kódy a pomocí SQL skriptu vytvořil databázi pro projekt Quiz. Zároveň jsem si nainstaloval všechny potřebné nástroje pro vývoj této aplikace, včetně správy databáze a SVN.

4.2 Analýza aplikace

Aplikace Quiz byla od začátku vyvíjena jako webová aplikace, která bude běžet na linuxovém serveru, a uživatelé k ní budou přistupovat pomocí prohlížeče v rámci intranetu. Na začátku vývoje tohoto projektu bylo rozhodnuto o tom, že bude použit webový framework Vaadin. Jak už jsem se zmínil, tak začátek mé práce spočíval v studování principu tohoto frameworku. Naučit se efektivně používat tento framework se mi nezdálo příliš složité, a to hlavně díky velice dobré podpoře, které Vaadin poskytuje na svých stránkách (www.vaadin.com). Na těchto stránkách lze najít jednotlivé funkční komponenty a také jednotlivé dema složené z více komponent. Aplikace byla navíc již skoro funkční, tím pádem nebylo zas tak těžké, zjistit jakým způsobem framework Vaadin pracuje.

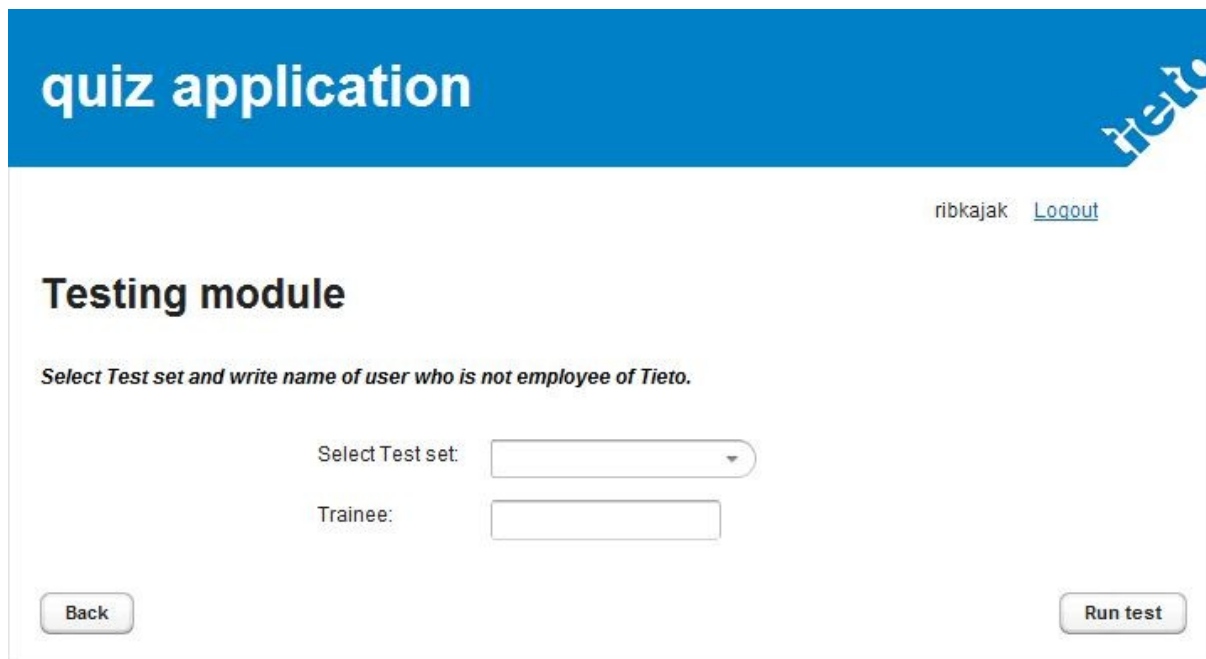
Úvodní obrazovku celé aplikace tvoří trojice tlačítek (k této aplikaci se uživatel dostane po přihlášení a ověření hesla vůči LDAPu), které slouží jako hlavní rozcestník celé aplikace. Je to tlačítko „Run test“, které slouží pro spuštění testu. Tlačítko „View result“, kterým lze zobrazit kompletní výsledky testů. A poslední tlačítko je tlačítko „Administration“, které slouží pro administraci aplikace. Uživatel, který se přihlásí do systému, má jednu ze čtyř rolí Super Admin, Admin, HR a User. O rolích v systému si více povíme v kapitole **4.2.4 Role v systému**. Podle toho, jaké má uživatel práva, se mění i úvodní stránka. Pokud se uživatel přihlásí jako Super Admin či Admin, má k dispozici všechna tři tlačítka, může tedy děla prakticky cokoliv. Uživatel HR má k dispozici jen prohlížení výsledků a spuštění testu, tedy tlačítka „View Result“ a „Run test“. Oproti tomu uživatel, který je přihlášený jako User, má jen jedinou volbu a tou je „Run test“, kterým se spustí test.



Obrázek 1: Titulní stránka

Po kliknutí na tlačítko „Run test“ je uživateli v závislosti na jeho roli v systému nabídnuta množina testů, které má k dispozici. Super Admin a HR vidí všechny testy, které jsou dostupné. Naproti tomu User si může vybrat jen z určitých testů, podle toho v jakých grupách (skupinách) je členem. O skupinách více v kapitole **4.2.3.7 Groups**. Po dokončení testu, se výsledek uloží do databáze. Další tlačítko je „View result“, ke kterému, jak už jsem se zmínil, má přístup Super Admin, Admin a HR. Posledním tlačítkem je tlačítko „Administration“, kterým se Super Admin či Admin dostane do jádra celé aplikace, a tím je Administrátorské rozhraní.

4.2.1 Run test



quiz application

ribkajak [Logout](#)

Testing module

Select Test set and write name of user who is not employee of Tieto.

Select Test set:

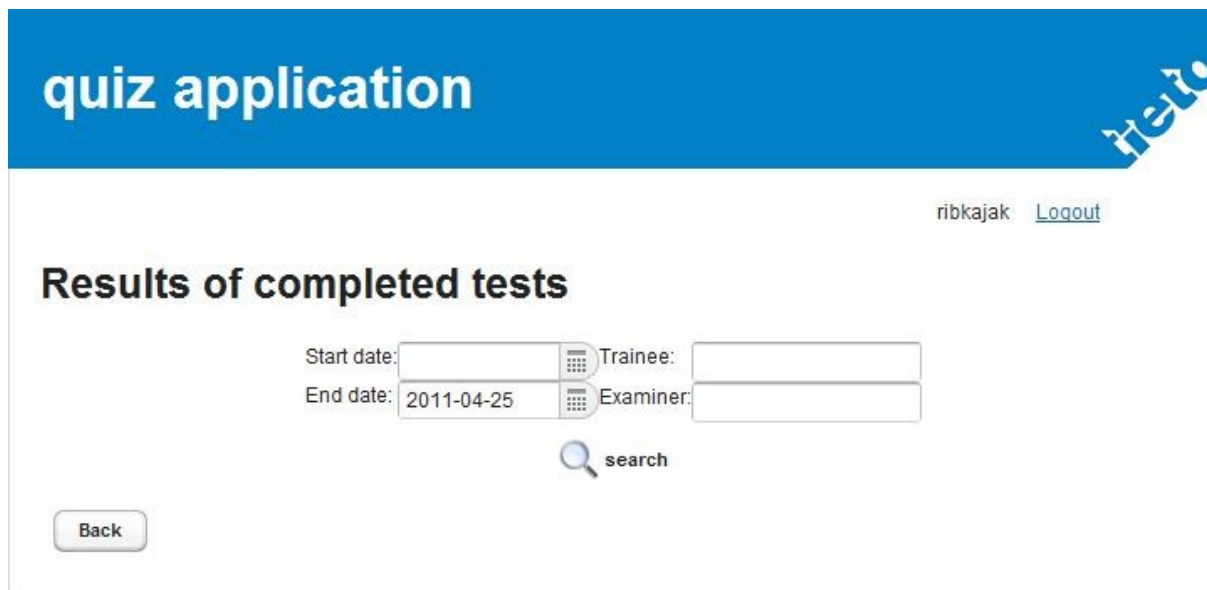
Trainee:

Back Run test

Obrázek 2: Run test

Jak jsem se již zmínil, výše po kliknutí na tlačítko „Run test“ se uživatel dostane do fáze, kdy si může vybrat jaký test, chce spustit. Hlavní omezení při tomto výběru u uživatelů s rolí User, a to z toho důvodu, aby uživatel nemohl nahlížet do jiných testů, než mu přísluší a nemohl tak např. vyžradit obsah testu někomu jinému více v kapitole **4.2.3.7 Groups**. Naproti tomu uživatelé s rolemi HR a Super Admin mohou vidět všechny dostupné testy.

4.2.2 View Result





quiz application


Hello

ribkajak [Logout](#)

Results of completed tests

Start date:  Trainee:

End date: 2011-04-25  Examiner:

 search

Back

Obrázek 3: View Result

K prohlížení všech výsledků se dostanou pouze uživatelé Super admin a HR. Admin se dostane pouze k výsledkům, které náleží jeho Grupě více **4.2.3.7Groups**. Toto omezení je zde opět z důvodu toho, aby obyčejný uživatel nemohl nahlížet do výsledků, ať už svých nebo v horším případě cizích testů. Po kliku na tlačítko „View result“ má uživatel několik možností, jak vyhledat a seřadit výsledky testů. Může vyhledat test podle určitého data, nebo podle rozmezí dvou dat. Možné je také vyhledávání podle zkoušejícího (Examiner), nebo podle toho, kdo byl zkoušen (Trainee).

4.2.3 Admin

quiz application

ribkajak [Logout](#)

Administration module

QuestionsTest setsTest resultsCategoriesLevelsUsersGroups

New

Search:

Choose categories *
Java
.NET

Choose levels *
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6

Find

ID	TEXT	TYPE	CATEGORY	LEVEL	VISIBLE	EDITABLE
1	What is the size...	single- choice	Java	Level 1	yes	no
2	A class cannot be...	single- choice	Java	Level 1	yes	no
3	Following code will result...	single- choice	Java	Level 1	yes	no
4	Following code will result...	single- choice	Java	Level 2	yes	no
5	Following code will result...	single- choice	Java	Level 2	yes	no
6	Following code will result...	single- choice	Java	Level 2	yes	no
7	A class can be...	single- choice	Java	Level 3	yes	no
8	Following code will result...	single- choice	Java	Level 3	yes	no
9	Following code will result...	single- choice	Java	Level 3	yes	no
10	Following code will result...	single- choice	Java	Level 4	yes	no

Back

Obrázek 4: Administration

Tlačítkem „Administration“ se dostaneme do srdce celé aplikace a tím je administrátorské rozhraní, přes které se celá aplikace ovládá. Administrace se skládá z několika záložek, které si popíšeme. Plnou administraci aplikace má k dispozici pouze uživatel Super Admin, který má právo plně využívat všechny funkcionality systému a provádět libovolné změny či úpravy. Omezenou funkcionalitu v administrátorském rozhraní má uživatel Admin, který vidí pouze záložky Questions, Test sets, Categories a Groups.

4.2.3.1 Questions

Záložka Questions slouží pro vytváření nových otázek pro testy. V záložce Questions má Super Admin možnost vidět všechny otázky, Admin může vidět pouze otázky, které sám vytvořil. Uživatel může vyhledávat podle několika kritérií a těmi jsou: Full textové vyhledávání (text, který je

součástí některé otázky), vyhledávání podle kategorie a vyhledávání podle levelů, výsledek je průnik všech těchto možností. Vyhledanou otázku může Super Admin či Admin editovat, ale to jen v případě, že již není využita v nějakém Test setu, o kterém se zmíním podrobněji v kapitole **4.2.3.2 Test sets**.

Další možností, kterou má Super Admin a Admin na této záložce, je vytvoření zcela nové otázky, učiní tak po kliku na tlačítko „New“. Po kliknutí na toto tlačítko musí uživatel vyplnit několik povinných údajů otázky. Těmi jsou Typ (jedna odpověď správná, více odpovědí správných, odpověď se skládá z textu), Text (to je samotný text otázky), Kategorii a Level (zvolí do jaké kategorie a levelu má daná otázka patřit) a nakonec musí přidat minimálně dvě odpovědi na tuto otázku, a to pomocí tlačítka „Add answer“. Výslednou otázku pak uloží do databáze po kliknutí na tlačítko „Save“.

4.2.3.2 Test sets

quiz application

ribkajak [Logout](#)

Administration module

Questions Test sets Test results Categories Levels Users Groups

New

ID	NAME	NUMBER OF QUESTIONS FROM LEVEL	CATEGORIES	LEVELS	FORMAT OF TEST SET
1	static1	3	[Java]	[Level 1, Level 2, Level 3, Level 4, Level 5, Level 6]	static
2	dynamic2	2	[.NET]	[Level 1, Level 2, Level 3, Level 4, Level 5, Level 6]	dynamic
3	dynamic3	2	[.NET]	[Level 1, Level 2, Level 3, Level 4, Level 5, Level 6]	dynamic

Back

Obrázek 5: Test sets

Další záložkou v Administraci je záložka Test sets. Test set si můžeme představit jako množinu připravených otázek, která je specifikována obtížností testu (Level), kategoriemi, ze kterých se test skládá (Categories) a počtem otázek z jednotlivého Levelu. Test set ještě nejsou fyzické otázky

připravené na test, ale pouze soubor pravidel, podle kterých se otázky budou náhodně vybírat poté, co uživatel v hlavní nabídce klikne na tlačítko „Run test“ a zvolí příslušný Test set.

I zde má možnost Super Admin či Admin jednotlivé Test sety editovat, nebo vytvořit zcela nový Test set. Při vytváření nového Test setu musí opět vyplnit několik povinných položek a těmi jsou: Name (jméno Test setu), Count of questions from each level (počet otázek z každé úrovně-Levelu), Format of Test set (statický či dynamický viz 4.4.2 **Dynamické testování**) a z Multi Choisu vybere z jakých kategorií a Levelů se bude příslušný Test set skládat. Test set uloží do databáze klikem na tlačítko „Save“.

4.2.3.3 Test results

quiz application

ribkajak [Logout](#)

Administration module

Questions Test sets **Test results** Categories Levels Users Groups

ID	TRAINEE	EXAMINER	TEST SET	DATE	RESULT	FOMAT OF TEST
429	ribkajak	ribkajak	static1	2011-04-24	0	normal

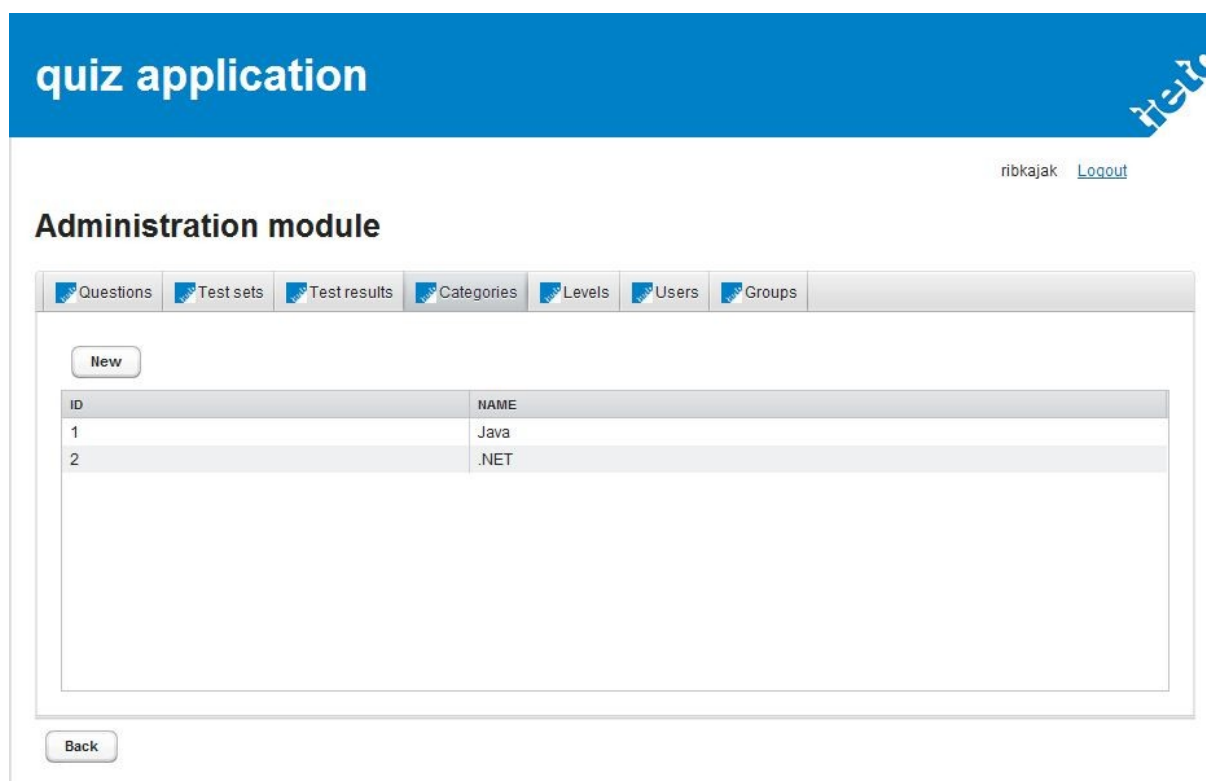
Back

Obrázek 6: Test results

K mazání, nikoliv k prohlížení, slouží další položka Administrátorského rozhraní a tou je položka Test results. Samotné prohlížení výsledků je dostupné po kliknutí na tlačítko „View result“ v hlavní nabídce, zde jde pouze o mazání těchto výsledků, které již nejsou z nějakého důvodu potřeba.

Super Admin jednoduše klikne na daný Test result v tabulce a pote klikne na tlačítko „Delete“, čímž se Test result smaže.

4.2.3.4 Categories



Obrázek 7: Categories

Kategorii, do které jednotlivé otázky spadají, nastavíme v záložce Categories. Kategorii si můžeme představit jako samotný okruh otázek, pokud je jméno kategorie Java, znamená to, že všechny otázky z této kategorie se týkají pouze Javy. Zde je opět možnost nahlédnout na již vytvořené Kategorie, případně editovat jejich název nebo vytvořit zcela novou kategorii. Při vytváření nové kategorie musíme vyplnit jen jednu povinnou položku a tou je položka Name (jméno dané kategorie). Po kliku na tlačítko „Save“ dojde opět k uložení kategorie do databáze.

4.2.3.5 Levels

quiz application

ribkajak [Logout](#)

Administration module

Questions Test sets Test results Categories **Levels** Users Groups

ID	NAME
1	Level 1
2	Level 2
3	Level 3
4	Level 4
5	Level 5
6	Level 6

Back

Obrázek 8: Levels

Levels je pouze informativní záložka a nemá žádnou speciální funkcionalitu, Admin a HR zde pouze mohou vidět Levely (v naší aplikaci neměně zvoleny v rozmezí Level 1 až Level 7).

4.2.3.6 Users

quiz application

ribkajak [Logout](#)

Administration module

Questions Test sets Test results Categories Levels **Users** Groups

New

ID	LOGIN	ROLE
1	kriegpet	Super Admin
2	ribkajak	Super Admin
3	baraktom	Super Admin
4	kvasnond	Super Admin
5	kaisesar	Super Admin
6	hecckmar	Super Admin
7	vantutom	Super Admin
8	zitniond	Super Admin
9	schifjan	Super Admin
10	sekulbar	Super Admin

Back

Obrázek 9: Users

V záložce Users najdeme všechny uživatele aplikace a jejich role, pokud se do aplikace přihlásí nový uživatel a jeho login a heslo jsou autentizovány vůči LDAPu, je v databázi vytvořen jako uživatel s nejnižšími právy, tedy jako User.

Super Admin na této záložce může jednotlivé uživatele editovat, měnit jim práva, mazat nebo vytvářet nové uživatele. Při vytváření nových uživatelů je důležité, aby vytvořený (nebo také editovaný) uživatel byl korektně zadán a ověřen vůči LDAPu, v opačném případě jeho uložení do databáze, které se opět provede tlačítkem „Save“, není možné.

4.2.3.7 Groups

quiz application

nbkajak [Logout](#)

Administration module

QuestionsTest setsTest resultsCategoriesLevelsUsers**Groups**

GROUP ID	GROUP NAME
1	Programmers
2	Administrators
3	HR

new groupdelete group

USER ID	USER LOGIN	GROUP ADMIN
7	vantutom	<input type="checkbox"/>
2	nbkajak	<input type="checkbox"/>
4	kvasnond	<input type="checkbox"/>
8	zitniond	<input type="checkbox"/>

add userremove user

TEST ID	TEST SET
2	dynamic2

add test setremove test set

Back

Obrázek 10: Groups

Jednou z nejdůležitějších položek v administraci je záložka Groups. Po kliknutí na záložku Groups se nám zobrazí jednotlivé Grupy – Skupiny. Tyto Skupiny můžeme chápat jako přiřazení jednotlivých uživatelů k jednotlivým Test setům. Toto nastavení je důležité při spouštění testů, např. pokud uživatele Pepík přidáme do skupiny programátoři, ke které přiřadíme Test sety: Java a .Net, uvidí Pepík při výběru testu, pouze testy odpovídající jeho skupině tedy Java a .Net a nebude mít možnost nahlížet do jiných testů.

Zde má Super Admin a Admin opět možnost editovat jednotlivé skupiny, či vytvářet nové. Ať už při vytváření nebo při editaci musí vybrat dvě povinné položky a těmi jsou, jak jsem se již zmínil, jednotliví uživatelé a test sety.

Speciální vlastnost ve skupině je funkce Group admin, funkcí Group admin může Super admin či Admin pověřit některého z uživatelů, kteří jsou součástí skupiny. Tato funkce byla vytvořena k tomu, aby Super Admin, ale spíše Admin, nemusel sám plně zodpovídat za zprávu skupiny, ale mohl část svých práv přenést na některého z uživatelů.

4.2.4 Role v systému

Důležitým kritériem v systému, podle kterého se rozhoduje, zda má uživatel právo na tu či onu operaci, jsou role v systému. V aplikaci Quiz rozlišujeme čtyři následující role.

4.2.4.1 Super Admin

Super admin je jeden člověk (není podmínkou), který má veškerou moc nad systémem. Zodpovídá zejména za udělování práv jiným uživatelům a samotnou správou těchto uživatelů. Dohlíží nad během celého systému a může mazat nepotřebné výsledky.

4.2.4.2 Admin

Druhou nejvyšší rolí v systému je role Admin. Do této role, jak již bylo řečeno, může libovolného uživatele (ověřeného vůči LDAPu) přiřadit Super admin. Admin je role, která vznikla za účelem tvoření Otázek, Kategorí, Test setů a Skupin. Ve firmě se nejčastěji jedná o vedoucího pracovníka (manažera), který chce testovat své zaměstnance nebo nabírá nové. Jeho práva jsou omezeny tak, aby mohl spravovat jen své prostředky (otázky, kategorie...), ale nemohl zasahovat do prostředků jiných Adminů (např. managerů z jiných oddělení).

4.2.4.3 HR

Role HR byla vytvořena pro personální pracovníky, kteří se v 99 % případů účastní všech pohovorů s uchazeči o nové zaměstnání a často v prvních kolech pohovorů jsou jediní, kteří jsou přítomni při pohovoru. Uživatelé s tímto právem mají možnost spustit libovolný test, který je k dispozici a rovněž vidí všechny výsledky, které jsou k dispozici.

4.2.4.4 User

Nejnižší roli v systému je role User, tuto roli získá každý uživatel, který je autorizován vůči LDAPu. Takovýto uživatel má v systému pouze právo na spuštění testu a po ukončení toho testu je ze systému automaticky odhlášen.

4.3 Nová funkcionality

Hned po mém nástupu do Tieto Czech s.r.o. jsem začínal dostávat úkoly na naimplementování nové funkcionality aplikace. K vývoji a celkově prohlubování mých znalostí z Javy určitě přispívaly každotýdenní meetingy s našim nadřízeným a později i s testy. Tato sezení pro mě měla obrovský přínos a pokud jsem si např. nevěděl rady, mohl jsem svůj problém prodiskutovat jak s týmem, tak s nadřízeným a dozvěděl se jiný názor na daný problém. Další výhodou těchto meetingů bylo codereview s našim nadřízeným, který má na pozici Java developera několikaletou praxi. Naš nadřízený vždy zkontroloval kód nové funkcionality, řekli jsme si, co je špatně a co by mělo být jinak, ale také to, co se mi povedlo. Mohl jsem si tak odnést ponaučení z chyb, které jsem při programování vytvářel.

4.4 Vývoj aplikace

Vývoj aplikace probíhal na počítačích firmy Tieto Czech s.r.o., programování na vlastních strojích bylo z důvodu bezpečnosti zakázáno. Jak jsem již zmínil, jako základní vývojové prostředí, byl použit Netbeans, ve kterém jsem mohl poměrně jednoduše spravovat svou lokální databázi, ale také spravovat testovací databázi běžící na linuxovém systému centrálně. Neocenitelný nástroj na sdílení kódů a jeho aktualizaci v týmu byl SVN, který rovněž běžel na linuxovém serveru. Na lokálním počítači jsem používal server Apache Tomcat, ve kterém běžela webová aplikace nebo přímo server GlassFish, který je součástí Netbeans. V rámci rozdělení jednotlivých úkolů a reportování chyb v aplikaci sloužila aplikace Jira, která měla nemalý podíl na urychlení vývoje této aplikace. Součástí Jiry je i graf, ze kterého lze lehce vyčíst, jak projekt postupuje a zda tým pracuje, jak má.

Jak již bylo řečeno, ihned po nástupu do Tieto Czech s.r.o., jsem začal dostávat úkoly, které jsem se snažil, co nejlépe vykonávat. Jelikož jednotlivých úkolů bylo opravdu mnoho, v následujících kapitolách se zmíním o těch nejdůležitějších.

4.4.1 Export výsledků do Wordu a PDF

Jako úplně první úkol po nástupu do Tieto Czech s.r.o. a přiřazení k projektu Quiz, bylo exportování výsledků testů do Wordu a PDF. Tento úkol jsem dostal spolu s jedním studentem na praxi a rozdělili jsme si ho tak, že já budu exportovat výsledky do Wordu a on do PDF.

Požadavek byl takový, že výsledný export má mít možnost formátování textu a pokud možno barevné rozlišení. Ve výsledném dokumentu se měl objevit absolvovaný test spolu se jménem zkoušeného a zkoušejícího a celkové vyhodnocení úspěšnosti testu.

Po dlouhém hledání vhodných knihoven, které bylo brzděno neznalostí interních systému a celkovým seznamováním se s prostředím, jsem se rozhodl použít knihovnu: **org.apache.poi.xwpf.usermodel.XWPFDocument**.

Nikdy předtím jsem s touto knihovnou nepracoval, proto jsem se práci s ní musel naučit. Nicméně požadavky na výsledný export se mi podařilo dodržet podle zadání.

Vyskytl se však problém, se kterým jsem si dlouho nevěděl rady a společně s kolegou, který se mnou na exportu pracoval, jsme jej dlouho dobu řešili. Problém spočíval v tom, že exportovaný dokument se začal otvírat na straně serveru a ne uživateli v prohlížeči, jak jsme potřebovali (na tuto skutečnost jsme přišli až poté, co se nám server podařilo shodit, v důsledku zahlcení vyexportovaných dokumentů). Problém spočíval ve špatném zvolení Streamu a tento problém se nám podařilo vyřešit takto:

```
public void write(boolean b) throws ExporterException {

    try {
        this.setDocument();
        final ByteArrayOutputStream byteStream = new ByteArrayOutputStream();
```

```

doc.write(byteStream);
byteStream.flush();
byteStream.close();
StreamSource ss = new StreamSource() {

InputStream is = new ByteArrayInputStream(byteStream.toByteArray());

@Override
public InputStream getStream() {
return is;
}
};
String nameOfDocument;
if (test.getUserByUserId() == null) {
nameOfDocument = "Default";
} else {
nameOfDocument = test.getUserByUserId().toString();
}
sr = new StreamResource(ss, nameOfDocument + "_" +
test.getDate().toString() + ".docx",
result.getMainWindow().getMainWindow().getApplication());
if (b) {
result.getMainWindow().getMainWindow().open(sr);
}
} catch (FileNotFoundException ex) {
throw new ExporterException(ex);
} catch (URISyntaxException ex) {
throw new ExporterException(ex);
} catch (IOException ex) {
throw new ExporterException(ex);
} catch (InterruptedException ex) {
throw new ExporterException(ex);
} catch (Exception ex) {
throw new ExporterException(ex);
}
}
}

```

Vhodnou kombinací Streamu se nám podařilo „podstrčit“ prohlížeči výsledný dokument a docílit tak otevření Save dialogu.

Mým dalším úkolem bylo výsledný dokument odeslat mailem uživateli, který byl testován. Tento úkol nebyl nijak náročný, jen jsem se musel zorientovat v nastavení firemního SMTP serveru. K odesílání emailu jsem použil knihovnu **javax.mail.***. Menší problém jsem měl s odesláním přílohy, ale podařilo se mi jej vyřešit.

4.4.2 Dynamické testování

Další velkým zásahem do aplikace bylo dynamické testování uživatelů. Klasické statické testování uživatelů v naší aplikaci probíhá standardně, jako při vyplňování běžného papírového testu. Naše dynamické testování se mělo lišit v tom, že testovaný uživatel bude dostávat otázky od nejtěžší po nejtěžší (Level 1 až Level 7). Byl zadán tento scénář:

Situace 1:

Příčina: Uživatel odpoví na první i druhou otázku z daného levelu správně.

Důsledek: Uživatel postoupí do dalšího levelu.

Situace 2:

Příčina: Uživatel odpoví na první i druhou otázku z daného levelu špatně.

Důsledek: Uživateli se ukončí testování dané kategorie a postoupí do další, pokud je k dispozici, pokud ne, je mu ukončen test.

Situace 3:

Příčina: V každém dalším případě dostává uživatel další otázku.

Důsledek: Uživatel postoupí do dalšího levelu nebo se ukončí testování dané kategorie v závislosti na počtu úspěšných a neúspěšných odpovědí.

Tento systém byl navrhnut za účelem soutěživosti stávajících zaměstnanců a prohlubování jejich znalostí, jelikož šlo velice snadno zjistit, jak daleko se daný zaměstnanec dostal. Znamenalo to zavést ke stávajícímu algoritmu (statickému) ještě další algoritmus. Byly nutné změny v databázi respektive v hibernatu a musela být změněna nebo doplněna řada tříd. Implementování spočívalo v zaznamenávání stavů, v kterých se uživatel právě nachází (první otázka, jedna chyba, dvě chyby apod.). Další změnou oproti klasickému testování bylo, že uživatel neměl možnost se v testu vrátit zpět (to by znemožnilo objektivní dynamické testování).

Na tomto úkolu jsem pracoval spolu s jedním členem týmu a plné naimplementování a zařazení do systému nám trvalo asi 8 dnů čistého času.

4.4.3 Editace otázek

V aplikaci, kterou jsme převzali v rozdělané podobě, nebyla možnost editovat již vytvořené otázky, a to se ukázalo postupem času jako veliký nedostatek. Celá editace měla fungovat tak, že bude možné změnit všechny položky otázky. Na pohled velice jednoduchá záležitost se dosti zkomplikovala po zjištění, že všechny atributy otázky jsou odkazy na cizí klíče jiných tabulek. Proto v tomto případě byl potřeba poměrně rozsáhlý zásah do databáze a úprava Hibernate mapování.

Po vyřešení funkcionality, bylo mým úkolem naprogramovat grafickou podobu editace otázek. Tu jsem řešil díky možnosti frameworku Vaadin tím, že jsem použil Layouty (Vertikální i Horizontální) a umístil je do Gridu, čímž jsem na celou stránku rozložil potřebné prvky(Tlačítka, Labely, ChecBoxy atd.)

4.4.4 Autentifikace pomocí LDAP

Problémem, který bylo potřeba řešit, byl přístup uživatelů do této aplikace. Ve verzi, kterou jsme dostali rozpracovanou, byl přístup uživatelů do aplikace řešen tak, že bylo nutné uživatele ručně v databázi vytvořit a teprve potom měl právo se do aplikace přihlásit. Toto se ukázalo jako naprosto nevhodná volba, když zvážíme nasazení aplikace v rámci celého Tieto Czech s.r.o. (cca 19 000 zaměstnanců), je téměř nereálné takto aplikaci spravovat. Proto bylo nutné zavést jiný systém přístupu do databáze.

Pro tento přístup jsem zvolil postup, kdy uživatel zadává svůj login a heslo a pokud je v ověření vůči LDAPu úspěšné, je vrácena instance třídy user se všemi potřebnými atributy. Toho přístupu jsem v aplikaci využil ještě na jednom místě, kdy bylo potřeba zjistit, zda se zadaný login nachází v LDAPu, a to při vytváření nových uživatelů, kterou si nyní popíšeme.

4.4.5 Vytváření a editace Uživatelů

Pro potřebu Super admina, který potřeboval nějakým způsobem vytvářet nové uživatele, jsem dostal za úkol vytvořit záložku v administrátorském rozhraní, touto záložkou byla záložka Users. Než byla tato záložka vytvořena, probíhalo vytváření nových uživatelé také ručně, což postupem času a s nárůstem složitosti přestalo stačit. Podrobný popis funkčnosti této záložky se nachází v kapitole 4.2.3.6 Users.

4.5 Testování aplikace

Jak již jsem se zmínil po nasazení testerů na projekt Quiz se začal charakter mé práce měnit a začala oprava naimplementovaných chyb. Jelikož z původně jednoho jediného testera se počet testerů rozrostl na tři, byl počet reportovaných chyb mnohonásobně vyšší. Ve většině případů se jednalo o chyby, kdy uživatel zadával do aplikace nějaké data a očekával reakci. Např. při zadávání názvů Otázek nebylo pamatováno na přehlasované znaky nebo znaky s diakritikou, tato chyba „probublala“ až do databázové vrstvy, kde bylo zjištěno, že původní kódování *latin* je nevhodné a bylo nahrazeno kódováním *utf-8*. Podobného charakteru byla většina reportovaných chyb a některé mi daly opravdu zabrat.

Testování výsledných funkcionalit, které se našemu týmu podařilo naimplementovat a oprava, těch chybných, nám zabrala téměř celou půlku naší odborné praxe. Velice mě překvapilo, jakou váhu a roli má tetování ve vývoji.

5 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.

Do firmy Tieto Czech s.r.o. jsem, jak už bylo řečeno, nastoupil na pozici Java developera a z tohoto důvodu jsem musel mít aspoň základní znalosti programovacího jazyka Java. Jelikož jsem se s jazykem Java na střední škole vůbec nesetkal, všechny potřebné informace jsem získal díky předmětům vyučovaným na Vysoké škole báňské. Jako největší přínos v této oblasti považuji předmět JAT, který jsem absolvoval v 5 semestru bakalářského studia, tedy souběžně s praxí. Přínosem v oblasti úplných základů byly rovněž předměty UPR a PJ1, ve kterých jsem pochopil základní rysy jazyka Java.

Jako velice užitečné se ukázaly předměty věnující se problematice databází. Při práci s databází na projektu Quiz jsem nejvíce uplatnil znalosti z předmětů TZD, DAIS a TIS.

Rovněž mi velice pomohl předmět SWS, díky kterému jsem mohl snadněji kreslit UML digramy a rychleji tak řešit danou problematiku v rámci týmů.

V rámci mého studia jsem se také setkal se CSS a HTML, tyto znalosti jsem uplatnil při návrhu grafické podoby naší aplikace.

6 Znalosti či dovednosti scházející studentovi v průběhu odborné praxe

První technologie, se kterou jsem se po příchodu do Tieto Czech s.r.o. setkal, byl webový framework Vaadin. S tímto frameworkem jsem se nikdy před mou praxí nesetkal. Nicméně, jak už jsem uvedl, díky výborné dokumentaci a jednoduše citovatelným komponentám jsem se tím webovým frameworkem naučil pracovat docela snadno.

Další z frameworků, který byl pro mě novinkou, byl framework Hibernate. Po příchodu do Tieto Czech s.r.o. měla sice již aplikace hotovou datovou vrstvu, ale stále větší a složitější změny vedly i ke změně této vrstvy, proto jsem byl nucen tento framework nastudovat a začít používat.

Asi jako největší nedostatek během celé mé praxe, se ukázala neschopnost psát UNIT testy. Na tuto důležitou věc jsme byli upozorňováni, ať už od svých zkušenějších kolegů, nebo od našeho nadřízeného. Myslím si, že předmět, který by se věnoval problematice testování aplikací, by byl v praxi velice užitečný.

7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Myslím si, že vykonání odborné praxe, jako ekvivalent bakalářské práce, je velice dobrý nápad a jsem rád, že jsem tuto praxi absolvoval. Získal jsem v Tietu mnoho užitečných a cenných zkušeností, ale hlavně jsem si dokázal, že jsem schopný se začlenit do týmu a být jeho plnohodnotným členem. Aplikace, na které jsem spolu s dalšími studenty VŠB pracoval, se po vedení zkušební verze zalíbila několika managerům, kteří požádali o její používání. Tento zájem považuji jako nejlepší měřítko toho, jestli jsme praxi absolvovali úspěšně, či ne. Měl jsem možnost potkat spoustu zajímavých a především velmi zkušených lidí, od kterých jsem se toho spoustu naučil. Celkově bych ohodnotil tuto praxi jako úspěšnou a přínosnou. A pokud bych mohl, rád bych ji doporučil dalším studentům naší fakulty.

8 Seznam obrázků

Obrázek 1: Titulní stránka	6
Obrázek 2: Run test.....	7
Obrázek 3: View Result	8
Obrázek 4: Administration	9
Obrázek 5: Test sets	10
Obrázek 6: Test results	11
Obrázek 7: Categories.....	12
Obrázek 8: Levels	13
Obrázek 9: Users	14
Obrázek 10: Groups	15

9 Literatura

1. Apache maven. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-05-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Apache_maven>.
2. Vaadin. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-05-03]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Vaadin>>.
3. Mysql. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-05-03]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Mysql>>.
4. Glassfish. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-05-03]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Glassfish>>.
5. BAUER, Christian; KING, Gavin. *Hibernate in Action : A guide to the concepts and practise of object/relational mapping*. [s.l.] : [s.n.], 2005. 400 s. ISBN 1932394–15-X.
6. BLOCH, Joshua . *Java efektivně : 57 zásad softwarového experta*. [s.l.] : GRADA Publishing, 2002. 232 s. ISBN 80-247-0416-1.